

Output-sensitive algorithm for generating the flats of a matroid

Alberto Montina

Perimeter Institute for Theoretical Physics, 31 Caroline Street North,
Waterloo, Ontario N2L 2Y5, Canada

July 22, 2011

Abstract

We present an output-sensitive algorithm for generating the whole set of flats of a finite matroid. Given a procedure, P , that decides in S_P time steps if a set is independent, the time complexity of the algorithm is $O(N^2 M S_P)$, where N and M are the input and output size, respectively. In the case of vectorial matroids, a specific algorithm is reported whose time complexity is equal to $O(N^2 M d^2)$, d being the rank of the matroid. In some cases this algorithm can provide an efficient method for computing zonotopes in H -representation, given their representation in terms of Minkowski sum of known segments.

1 Introduction

A matroid is a structure, introduced by Whitney [1], providing an abstraction of the concept of independence that is common in different theories, such as linear algebra and graph theory. A finite matroid is defined as a pair $(\mathbf{W}, \mathcal{I})$, where \mathbf{W} is a finite set, called *ground set*, and \mathcal{I} is a collection of subsets of \mathbf{W} , called independent sets, satisfying the following three properties [2]:

1. The set \mathcal{I} is not empty;
2. If $\mathbf{A} \in \mathcal{I}$ and $\mathbf{B} \subseteq \mathbf{A}$, then $\mathbf{B} \in \mathcal{I}$ (*heredity*);
3. If $\mathbf{A}, \mathbf{B} \in \mathcal{I}$ and $|\mathbf{A}| < |\mathbf{B}|$, then there exists an element $b \in \mathbf{B} \setminus \mathbf{A}$ such that $\{b\} \cup \mathbf{A} \in \mathcal{I}$ (*augmentation property*),

$|\mathbf{C}|$ being the cardinality of set \mathbf{C} . With a slight abuse of notation, hereafter we will denote by $b \cup \mathbf{A}$ the union of two sets \mathbf{A} and $\{b\}$, the latter containing the single element b . In this paper we will refer often to the concrete example of vectorial matroids, where the elements of \mathbf{W} are vectors of a vector space and independent sets are the linearly independent subsets of \mathbf{W} .

We are interested in an efficient algorithm for computing flats of a matroid. Flats are subset of \mathbf{W} whose properties provide an alternative axiomatization of matroids. Ordered by inclusion, they form a geometric lattice [2, 3]. In the case of vectorial matroid, each flat with maximal rank and properly contained in \mathbf{W} (called *hyperplane*) can be associated with two facets of a zonotope up to translations. Zonotopes are polytopes, equivalently defined as Minkowski sum of segments or affine projections of cubes. They play an important role in several mathematical areas, such as hyperplane arrangements, box splines and partition functions [4]. They could turn to be useful also in some problems of quantum information.

In general the number of flats grows exponentially in $|\mathbf{W}|$. Thus, their computation has an exponential time complexity. However, in many practical problems the matroid has special properties that considerably reduce the output size with respect to the general case. For example, this occurs if the cardinality of some dependent subset of \mathbf{W} is smaller than or equal to the matroid rank. An algorithm whose running time depends only on the input size does not take advantage of these special structures and its complexity is exponential in any case. Conversely, an output-sensitive algorithm, whose running time depends on both the input and output size, may require much less resources in cases of reduced output size. We will consider the extended notion of polynomial complexity that accounts for this output sensitivity. An algorithm is polynomial if its running time is polynomial in both the input and output size.

In this paper we will present an output-sensitive algorithm for computing all the flats of a matroid. Assuming that there is a procedure that decides in S_P steps if a set is independent, the complexity of evaluating the flats is $O(N^2MS_P)$, where M is the number of flats and N is the cardinality of \mathbf{W} . The linearity in the number of flats is the significant feature that makes the algorithm output-sensitive. In the concrete case of vectorial matroid, the procedure P can be given for example by an algorithm that evaluates the rank of matrices. The overall complexity of evaluating the flats is $O(N^2Md^3)$. We will also provide a specific optimization that requires an insignificant increase of computation space and reduces the time complexity to $O(N^2Md^2)$. Since the facets of a zonotope are identified by flats of a vectorial matroid up to translations, in some cases our algorithm can provide an efficient method for evaluating the H -representation of this particular polytope. The paper is organized as follows. In Sec. 2 we give some definitions, like *rank*, *basis*, *closure* and *flat*. We show that each flat can be represented through its bases. In sec. 3, we introduce a total order in the power set of the ground set \mathbf{W} and use it to select a representative basis for each flat. Some properties of this basis are proved and then used in Sec. 4, where we present the algorithm for generating the whole set of flats. In Sec. 5, we show that our algorithm can provide in some cases an efficient method for evaluating the H -representation of a zonotope, given its representation in terms of Minkowski sum of known segments.

2 Definitions and basic properties

In order to define flats, it is useful to introduce some concepts, such as basis, rank and closure. A *basis* is a maximal independent set, that is, an independent set that is not properly contained in an independent set. The last axiom of matroid theory implies that all the bases have the same cardinality, which is called *rank* of the matroid. Given a matroid $(\mathbf{W}, \mathcal{I})$ and any subset \mathbf{U} of \mathbf{W} , let \mathcal{J} be the collection of subsets of \mathbf{U} that are in \mathcal{I} , then the pair $(\mathbf{U}, \mathcal{J})$ is a matroid. It is called the *restriction* of $(\mathbf{W}, \mathcal{I})$ to \mathbf{U} [2]. Thus, each $\mathbf{U} \subseteq \mathbf{W}$ has a rank, denoted by $r(\mathbf{U})$, and a set of bases.

The *closure* of a set $\mathbf{U} \subseteq \mathbf{W}$, indicated with $\text{cl}(\mathbf{U})$, contains the elements $w \in \mathbf{W}$ such that \mathbf{U} and $\mathbf{U} \cup w$ have the same rank, that is,

$$\text{cl}(\mathbf{U}) := \{w \in \mathbf{W} | r(\mathbf{U} \cup w) = r(\mathbf{U})\}. \quad (1)$$

The closure operator is idempotent, that is,

$$\text{cl}(\text{cl}(\mathbf{U})) = \text{cl}(\mathbf{U}), \quad \forall \mathbf{U} \subseteq \mathbf{W}. \quad (2)$$

A *flat* is a set $\mathbf{F} \subseteq \mathbf{W}$ that is equal to its closure, that is,

$$\mathbf{F} \text{ is a flat} \Leftrightarrow \mathbf{F} = \text{cl}(\mathbf{F}). \quad (3)$$

A flat of rank n is called n -flat. The $(d-1)$ -flat of a matroid of rank d is called *hyperplane*. Flats are analogous to vector subspaces. Indeed, in the case of vectorial matroids, it is possible to identify the flats with the subspaces linearly spanned by the vectors in the flats. In particular, n -flats correspond to n -dimensional subspaces.

By the idempotence property (2) we have that

$$\forall \mathbf{X} \subseteq \mathbf{W}, \quad \text{cl}(\mathbf{X}) \text{ is a flat.} \quad (4)$$

Trivially, the closure operator, with the sets of flats as codomain, is a surjective function, thus it is possible to represent every flat \mathbf{F} through a subset whose closure is \mathbf{F} . The minimal sets representing a flat \mathbf{F} are independent sets with cardinality equal to $r(\mathbf{F})$, as implied by the followings.

Lemma 2.1 *If $\mathbf{X} \subseteq \mathbf{Y}$ and $r(\mathbf{X}) = r(\mathbf{Y})$, then $\text{cl}(\mathbf{X}) = \text{cl}(\mathbf{Y})$.*

Lemma 2.2 $\forall \mathbf{X} \quad r(\text{cl}(\mathbf{X})) = r(\mathbf{X})$.

As a direct consequence, we have the following lemma.

Lemma 2.3 *The sets with minimal cardinality that represent a flat \mathbf{F} are all the bases of \mathbf{F} .*

Proof. Let \mathbf{S} be a basis of \mathbf{F} , thus, by definition of basis and rank, $\mathbf{S} \subseteq \mathbf{F}$ and $r(\mathbf{S}) = r(\mathbf{F})$. By Lemma 2.1 and definition of flat we have that $\text{cl}(\mathbf{S}) = \text{cl}(\mathbf{F}) = \mathbf{F}$, that is, \mathbf{S} represents the flat \mathbf{F} . This set is also minimal because of Lemma 2.2. Indeed, if \mathbf{R} represents the flat \mathbf{F} , then by definition $\text{cl}(\mathbf{R}) = \mathbf{F}$. By Lemma 2.2, this implies that $r(\mathbf{R}) = r(\mathbf{F})$, that is, the cardinality of \mathbf{R} is not smaller than the cardinality of a basis of \mathbf{F} .

Thus, flats can also be identified with the class of their bases, just as in linear algebra a set of k independent vectors identifies a k -dimensional linear subspace. This provides a simplification in the representation of flats, since it is not necessary to enumerate the whole set of its elements. The mapping from the bases to the flats is surjective, but in general is not bijective. In the next section, we introduce a rule for selecting a representative basis that will turn to be fundamental for developing our algorithm.

3 Denoting flats through representative bases

To select a representative basis of a flat, we introduce a total order on the subsets of \mathbf{W} and associate each flat with its *first* basis (“first” with respect to the total order). The order is defined as follows.

Given a matroid $(\mathbf{W}, \mathcal{I})$, we order the elements of \mathbf{W} by appending an integer $i \in [1, N]$ to each element $w_i \in \mathbf{W}$, N being the cardinality of \mathbf{W} . Then we represent a subset $\mathbf{U} \subseteq \mathbf{W}$ through a N -digit binary number by setting the i -th digit equal to 1(0) if w_i is (not) an element of \mathbf{U} , for every $i \in [1, N]$. In other words, given any collection $\{b_1, \dots, b_k\}$ of indices with $b_{n-1} < b_n$, we label the subset $\mathbf{X} = \{w_{b_1}, w_{b_2}, \dots, w_{b_k}\} \subseteq \mathbf{W}$ with the number

$$L(\mathbf{X}) = \sum_{n=1}^k 2^{b_n - 1}. \quad (5)$$

For example,

$$L(\{w_1, w_3, w_7\}) = 1000101_2,$$

which is equal to 69_{10} in decimal basis. We call the most significant nonzero bit of a binary number b *leading* digit of b .

By attaching the label L to each subset of \mathbf{W} , we have introduced a total order in the power set of \mathbf{W} . We associate each flat \mathbf{F} with the basis that has the smallest label L , which we call *pointer* of \mathbf{F} . In particular an n -pointer is the pointer of an n -flat. The pointer of \mathbf{F} will be indicated with $p(\mathbf{F})$. It is important to distinguish the pointer of a flat from its label L . The pointer of \mathbf{F} is the label of its first basis, which in general is different from the label of \mathbf{F} , unless \mathbf{F} is independent. In binary representation, the number of non-zero digits of the pointer and label of a flat \mathbf{F} is equal to $r(\mathbf{F})$ and $|\mathbf{F}|$, respectively.

Theorem 3.1 *Let p_0 be the pointer of a n -flat in binary representation, then the number s obtained from p_0 by replacing the leading digit with 0 is the pointer of an $(n - 1)$ -flat.*

In order to prove it, we need the following property.

Lemma 3.2 *Let \mathbf{I}_1 and \mathbf{I}_2 be two independent subset of \mathbf{W} such that $\text{cl}(\mathbf{I}_1) = \text{cl}(\mathbf{I}_2)$, then, for every $w \in \mathbf{W}$, $\text{cl}(\mathbf{I}_1 \cup w) = \text{cl}(\mathbf{I}_2 \cup w)$.*

Proof. If $w \in \text{cl}(\mathbf{I}_1) = \text{cl}(\mathbf{I}_2)$, then the lemma is a direct consequence of the implication $w \in \text{cl}(\mathbf{X}) \Rightarrow \text{cl}(\mathbf{X}) = \text{cl}(\mathbf{X} \cup w)$, which can be easily obtained from lemma 2.1. Thus, let us consider the case $w \notin \text{cl}(\mathbf{I}_1) = \text{cl}(\mathbf{I}_2)$, that is, we assume that $\mathbf{I}_1 \cup w$ and $\mathbf{I}_2 \cup w$ are independent. We have to prove that if an element $v \notin \text{cl}(\mathbf{I}_1 \cup w)$, then $v \notin \text{cl}(\mathbf{I}_2 \cup w)$ and vice versa. Suppose that $v \notin \text{cl}(\mathbf{I}_1 \cup w)$, then $\mathbf{I}_1 \cup w \cup v$ is independent. Since also $\mathbf{I}_2 \cup w$ is independent and $|\mathbf{I}_1 \cup w \cup v| > |\mathbf{I}_2 \cup w|$, by axiom 3 of matroid theory (augmentation property) there is an element $b \in \mathbf{I}_2 \cup w \cup v$ such that $\mathbf{I}_2 \cup w \cup b$ is independent. Clearly, b is not in \mathbf{I}_1 , since $\text{cl}(\mathbf{I}_1) = \text{cl}(\mathbf{I}_2)$ and $b \notin \mathbf{I}_2$. Furthermore, b cannot be equal to w , thus $b = v$. This implies that $\mathbf{I}_2 \cup w \cup v$ is independent, that is, $v \notin \text{cl}(\mathbf{I}_2 \cup w)$. Also the inverse implication is true. Thus, every element that is not in $\text{cl}(\mathbf{I}_1 \cup w)$ is not in $\text{cl}(\mathbf{I}_2 \cup w)$ and vice versa, that is, $\text{cl}(\mathbf{I}_1 \cup w)$ and $\text{cl}(\mathbf{I}_2 \cup w)$ are equal. \square

Proof of Theorem 3.1. Let \mathbf{I}_1 be the independent set with label s . Suppose that s is not a pointer, thus there is an independent set \mathbf{I}_2 with $L(\mathbf{I}_2) < L(\mathbf{I}_1)$ such that $\text{cl}(\mathbf{I}_2) = \text{cl}(\mathbf{I}_1)$. Let k be the position of the leading digit of p_0 , then, by definition of s and \mathbf{I}_1 , $\mathbf{I}_1 \cup w_k$ is the independent set pointed to by p_0 . By lemma 3.2, both $\mathbf{I}_1 \cup w_k$ and $\mathbf{I}_2 \cup w_k$ are bases of the same flat and furthermore $L(\mathbf{I}_2 \cup w_k) < L(\mathbf{I}_1 \cup w_k)$,

since $L(\mathbf{I}_2) < L(\mathbf{I}_1)$, but this is impossible because $p_0 = L(\mathbf{I}_1 \cup w_k)$ is a pointer. \square

This lemma implies that each $(n+1)$ -pointer can be generated from some n -pointer by setting one of the digits at the left of the leading digit equal to 1. For example, if $a = 10010011_2$ is a 4-pointer, then the number $b = 00010011_2$ is a 3-pointer for Theorem 3.1. The 4-pointer a is generated from the 3-pointer b by replacing the 8-th zero digit of b with 1. Thus, given a collection of n -pointers, this procedure of replacement generates a set of labels that contains the set of all the $(n+1)$ -pointers. In general the inclusion is strict, that is, the procedure of adding a bit 1 to an n -pointer does not necessarily give an $(n+1)$ -pointer and we need a criterion for discarding labels that are not pointers.

Theorem 3.3 *Let ‘ s ’ and ‘ δ ’ be the label of an subset \mathbf{X} and the position of the leading digit of s , respectively. Let \mathbf{Y}_i be the set obtained from \mathbf{X} by removing the element $w_j \in \mathbf{X}$ with $j > i$. The integer s is a pointer if and only if \mathbf{X} is independent and, for every $w_k \notin \mathbf{X}$ with $k < \delta$, $w_k \notin \text{cl}(\mathbf{X})$ or $w_k \in \text{cl}(\mathbf{Y}_k)$.*

Note that \mathbf{Y}_i and \mathbf{Y}_j are not necessarily different if $i \neq j$. In order to prove this lemma, we need three properties. The first one is known as the *Mac Lane-Steinitz exchange property* [2].

Lemma 3.4 *Given a subset $\mathbf{X} \subseteq \mathbf{W}$ and an element $w \in \mathbf{X}$, if $v \in \text{cl}(\mathbf{X})$ and $v \notin \text{cl}(\mathbf{X} \setminus w)$, then $\text{cl}(\mathbf{X} \setminus w \cup v) = \text{cl}(\mathbf{X})$.*

The second one is the *basis exchange property* [2].

Lemma 3.5 *If \mathbf{B}_1 and \mathbf{B}_2 are two bases of a subset of \mathbf{W} and $w \in \mathbf{B}_1$, then there is an element $v \in \mathbf{B}_2 \setminus \mathbf{B}_1$ such that $\mathbf{B}_1 \setminus w \cup v$ is a basis.*

Finally, the last property is a consequence of the augmentation axiom.

Lemma 3.6 *Let \mathbf{Y} be a subset of $\mathbf{X} \in \mathcal{I}$. If $w \notin \text{cl}(\mathbf{Y})$, then there is an element $v \in \mathbf{X} \setminus \mathbf{Y}$ such that $w \notin \text{cl}(\mathbf{X} \setminus v)$.*

Before proving theorem 3.3, let us prove the last property.

Proof of lemma 3.6. First, we assume that $w \in \mathbf{X}$. It is clear that if $v = w$ then $w \notin \text{cl}(\mathbf{X} \setminus v)$, since \mathbf{X} is independent. Furthermore $v \in \mathbf{X} \setminus \mathbf{Y}$, since $w \in \mathbf{X}$ and $w \notin \mathbf{Y}$, and the conclusion of the lemma is proved. Now we assume that $w \notin \mathbf{X}$. For the augmentation property, it is possible to construct an independent set by adding $|\mathbf{X}| - |\mathbf{Y} \cup w|$ elements in $\mathbf{X} \setminus \mathbf{Y}$ to $\mathbf{Y} \cup w$. The obtained set is equal to $\mathbf{X} \cup w$ minus some element $v \in \mathbf{X} \setminus \mathbf{Y}$. Thus, $\mathbf{X} \setminus v \cup w$ is independent, that is, $w \notin \text{cl}(\mathbf{X} \setminus v)$. \square

Proof of theorem 3.3. First we prove one direction of the implication and assume that s is the pointer of a flat \mathbf{F} . By definition \mathbf{X} is independent. Suppose that the other part of the conclusion is false, thus there is an element $w_k \notin \mathbf{X}$ with $k < \delta$ such that $w_k \in \text{cl}(\mathbf{X})$ and $w_k \notin \text{cl}(\mathbf{Y}_k)$. Last condition and lemma 3.6 imply that there is an element $w_l \in \mathbf{X} \setminus \mathbf{Y}_k$ such that $w_k \notin \text{cl}(\mathbf{X} \setminus w_l)$. By definition of \mathbf{Y}_k we have that $l > k$. Thus, since $w_k \in \text{cl}(\mathbf{X})$ and $w_k \notin \text{cl}(\mathbf{X} \setminus w_l)$, by lemma 3.4 we have that $\text{cl}(\mathbf{X} \setminus w_l \cup w_k) = \text{cl}(\mathbf{X})$, that is, $\mathbf{X} \setminus w_l \cup w_k$ is a basis of \mathbf{F} , but this is impossible because $L(\mathbf{X} \setminus w_l \cup w_k) < s$ (l is greater than k) and s is a pointer, thus one direction of the implication is proved.

Let us prove the other direction. Suppose that s is not a pointer, then there is a number $s_1 < s$ such that $L^{-1}(s_1) \equiv \bar{\mathbf{X}}$ and $L^{-1}(s) = \mathbf{X}$ are bases of the same flat \mathbf{F} . Let w_l be the element with largest subscript l such that $w_l \in \mathbf{X}$ and $w_l \notin \bar{\mathbf{X}}$. Since $s_1 < s$, this element exists. By lemma 3.5 there is an element $w_k \in \bar{\mathbf{X}} \setminus \mathbf{X}$, such that $\mathbf{X} \setminus w_l \cup w_k$ is independent, that is, $w_k \notin \text{cl}(\mathbf{X} \setminus w_l)$. The elements with subscript larger than l are in \mathbf{X} if and only if are in $\bar{\mathbf{X}}$. Since $w_k \notin \mathbf{X}$ and $w_k \in \bar{\mathbf{X}}$, then k cannot be larger than l , thus we have that $k < l$. Because of this inequality, \mathbf{Y}_k is a subset of $\mathbf{X} \setminus w_l$. The relations $\mathbf{Y}_k \subseteq \mathbf{X} \setminus w_l$ and $w_k \notin \text{cl}(\mathbf{X} \setminus w_l)$ imply that $w_k \notin \text{cl}(\mathbf{Y}_k)$. It is also clear that $w_k \in \text{cl}(\mathbf{X})$, since w_k is in $\bar{\mathbf{X}}$ and $\text{cl}(\bar{\mathbf{X}}) = \text{cl}(\mathbf{X})$. \square

4 Algorithm for generating the flats

Theorems 3.1 and 3.3 are the two key ingredients of our algorithm for calculating the flats of a matroid. The idea is generating recursively the i -pointers from the lower-dimensional $(i-1)$ -pointers. The flats are then generated from their pointers. More precisely, a set of labels is generated from each $(i-1)$ -pointer by setting

one of the digits at the left of the leading digit equal to 1. If l is the position of the leading digit of an $(i-1)$ -pointer and N is the number of elements in \mathbf{W} , then a $(i-1)$ -pointer generates $N-l$ labels. The set of labels generated from all the $(i-1)$ -pointers contains the whole set of i -pointers. Theorem 3.3 provides an efficient method for discarding labels that are not pointers.

Since the structure of flats is unaffected by the presence of loops and parallel elements, we will assume without loss of generality that they are absent, that is, we assume that the matroid is simple [2]. Loops are dependent subsets of \mathbf{W} with cardinality equal to 1. In the case of vectorial matroids, a loop is a zero vector. Parallel elements are pairwise dependent vectors. Denoting by N and d the cardinality and the rank of the matroid, respectively, the algorithm for generating the pointers is as follows.

Input: The set of N 1-pointers {the elements in \mathbf{W} }

```

1: Set  $\bar{M}$  equal to number of 1-pointers.  $\{:= N\}$ 
2: for  $i = 2, \dots, d-1$  do
3:   for  $j = 1, \dots, \bar{M}$  do
4:     Set  $l$  equal to the position of the leading digit of the  $j$ -th  $(i-1)$ -pointer.
5:     for  $\delta = l+1, \dots, N$  do
6:       Generate from  $j$ -th  $(i-1)$ -pointer a label  $s$  by setting the  $\delta$ -th digit equal to 1.
7:       if  $s$  is a pointer {This is checked through theorem 3.3} then
8:         Store  $s$  as a new  $i$ -pointer.
9:       end if
10:    end for
11:  end for
12:  Set  $\bar{M}$  equal to the number of  $i$ -pointers.
13: end for

```

Output: The whole set of pointers

The check at line 7 is performed through theorem 3.3. Thus, it requires to verify that the set \mathbf{X} pointed to by s is independent and, for every $w_k \notin \mathbf{X}$ with $k < \delta$, the set $\mathbf{X} \cup w_k$ is independent or $\mathbf{Y}_k \cup w_k$ is dependent. Given a procedure, P , that decides with S_P steps if a set is independent, the time complexity of the check at line 7 is $O(NS_P)$. The overall time complexity of the algorithm is $O(N^2MS_P)$, where M is the total number of flats, that is, $M = \sum_{i=1}^{d-1} M_i$, M_i being the number of i -flats.

In the specific case of a vectorial matroid, the procedure P can be provided for example by a routine that evaluates the rank of matrices. Indeed any set $\mathbf{X} \subseteq \mathbf{W}$ of a vectorial matroid can be seen as a $d \times |\mathbf{X}|$ matrix (called column matroid), the columns being the elements of \mathbf{X} . The rank of \mathbf{X} is the rank of the matrix. The set is independent if the rank is equal to $|\mathbf{X}|$. The time complexity of evaluating the flats by using this routine is $O(N^2Md^3)$. In this scheme the rank of the matrices is evaluated without taking advantage of the similarity of their structure. Indeed it is possible to reduce the time complexity by a slight increase of the computational space that exploits this similarity. Let p_0 and \mathbf{Z} be an $(i-1)$ -pointer and its corresponding independent set. l is the position of the leading digit of p_0 . $N-l$ labels are generated from p_0 by setting the δ -th digit equal to 1, where δ is an integer ranging between $l+1$ and N (see line 5 in the algorithm). We denote by $\mathbf{X}^{(\delta)}$ the sets associated with the generated labels. In line 7 of the algorithm, first we have to verify that $\mathbf{X}^{(\delta)}$ is independent for each δ . Since sets with different values of δ differ in one element and share the subset \mathbf{Z} , the best strategy to decide if the sets $\mathbf{X}^{(\delta)}$ are independent is, first, reducing \mathbf{Z} to the row echelon form using row operations and, then, performing the same row operations on the added column in each $\mathbf{X}^{(\delta)}$. The corresponding time complexity is $O(d^2N)$, taking into account that $d \leq N$. A similar strategy can be used also in checking the independence of $\mathbf{X} \cup w_k$ and the dependence of $\mathbf{Y}_k \cup w_k$. In this way it is possible to reduce the overall complexity of generating the flats to $O(N^2Md^2)$.

5 Minkowski sum of segments: zonotope

The algorithm for the computation of flats can be useful in some cases for calculating the H -representation of a zonotope when it is represented as Minkowski sum of known segments. The Minkowski sum of two sets

A and B in a vector space is the set obtained by adding every vector of A to every vector of B , that is,

$$A + B = \{\vec{a} + \vec{b} | \vec{a} \in A, \vec{b} \in B\}. \quad (6)$$

The zonotope is a polytope defined as the Minkowski sum of segments. Up to a translation, it is the set of vectors

$$\vec{v} = \sum_{k=0}^M \lambda_k \vec{w}_k, \quad (7)$$

where $\lambda_k \in [0 : 1]$. Each vector \vec{w}_k and the zero vector $\vec{0}$ are the two vertices of each segment summed up. The set $\mathbf{W} = \{\vec{w}_k | k \in [1 : M]\}$ is the ground set of a vectorial matroid, whose independent sets are the sets of linearly independent vectors. Let d be the dimension of the zonotope, that is, the maximal number of independent vectors in \mathbf{W} . Without loss of generality, we assume that d is also the dimension of the vector space.

Each facet is parallel to a $(d - 1)$ -flat of the matroid, thus its normal vector is orthogonal to any basis of the $(d - 1)$ -flat. The computation of each normal vector by a set of $d - 1$ vectors has a complexity that scales like d^3 . We denote a normal vector with \vec{n}_i , where the subscript i is an integer that goes from 1 to M_{d-1} , M_{d-1} being the number of $(d - 1)$ -flats. It can be proved that for each vector \vec{n}_i there exist two facets defined by the inequalities

$$\vec{n}_i \cdot \vec{x} \leq \sum_k \theta(\vec{n}_i \cdot \vec{w}_k) \vec{n}_i \cdot \vec{w}_k \quad (8)$$

and

$$\vec{n}_i \cdot \vec{x} \geq - \sum_k \theta(-\vec{n}_i \cdot \vec{w}_k) \vec{n}_i \cdot \vec{w}_k, \quad (9)$$

where $\theta(x)$ is the step function $\theta(x > 0) = 1$, $\theta(x < 0) = 0$. These inequalities define the zonotope in H -representation. Thus, the computation of the zonotope in H -representation is achieved by evaluating the $(d - 1)$ -flats of the matroid \mathbf{W} .

The output in this problem is the set of half-planes, thus its size is M_{d-1} . In general our algorithm for the computation of the flats does not allow us to solve this problem in polynomial time with respect to the output size, since the algorithm presented in the previous section is linear in the total number of flats M and in the worst case M could be exponentially greater than M_{d-1} . However, in many practical problems M can scale linearly in M_{d-1} and the input size. Suppose for example that the vectors \vec{w}_k are in general position and the matroid rank d is smaller than $N/2$. The number of k -flats is $\frac{N!}{k!(N-k)!}$ and grows monotonically in $k (< d - 1)$. This implies that M scales at most like dM_{d-1} . This linear scaling can be present also in the case of special structure for which an output-sensitive algorithm provides an advantage. It is worthwhile to note that the best algorithm for the evaluation of a zonotope in H -representation has a complexity that is quadratic in the output size in any case [5]. In the subclass of problems where the number of overall flats is a linear function of the number of hyperplanes, our method for the generation of the zonotope in H -representation is linear in the output size. An open question is determining how much large is this subclass

Acknowledgments

The Author acknowledges useful discussions with Komei Fukuda, Hans Raj Tiwary and Erik Schnetter. Research at Perimeter Institute for Theoretical Physics is supported in part by the Government of Canada through NSERC and by the Province of Ontario through MRI.

References

- [1] Hassler Whitney, *On the abstract properties of linear dependence*, Am. J. Math. **57** (1935), 509.
- [2] James G. Oxley, *Matroid theory*, Oxford University Press, Oxford (1992).
- [3] G. Birkhoff, *Lattice Theory*, American Mathematical Society, Providence (1984).

- [4] C. De Concini, C. Procesi, *Topics in hyperplane arrangements, polytopes and boxsplines*, Springer (2010).
- [5] P. D. Seymour, *A note on Hyperplane Generation*, J. Comb. Theory B **61** (1994), 88.